

University of Hull
Department of Computer Science

Writing Games with Pygame

Vsn. 1.0 Rob Miles 2014

Week 1: Getting Started with Pygame

These notes describe the practical elements of the course. They are to be used in conjunction with the slide decks.

Practical Break 1: Getting a Drawing Surface

In this break we are going to get a drawing surface.

Getting started

We are going to use the shell to start with, just to make sure that pygame is installed and that we can get hold of a surface to draw on.



Before you go any further; perform the following:

1. Log into your system with the username and password that you know and love from the past.
2. Start the Idle environment.

Creating a Pygame drawing surface

To get started we will need to initialise the Pygame system and then begin drawing with it. The statements to do this are as follows:

```
import pygame
pygame.init()
size = (800,600)
surface = pygame.display.set_mode(size)
pygame.display.set_caption("My awesome game")
```



Before you go any further; perform the following:

3. Type each statement above in turn into the shell. Note what happens after the fourth line. You should see something appear on the screen.
4. See if you can work out how you would change the size of the display to 1024x768 pixels. Make the changes and see if they work.

When you have done this you can relax in the knowledge of a job well done.

Practical Break 2: Drawing a line and some other stuff

In this break we are going to draw some lines on the surface that we just got hold of. We are going to start by drawing a line:

```
pygame.draw.line(surface, (255,0,0), (0,0), (50,20))
pygame.display.flip()
```

The two statements above will draw a red line down the screen.



Before you go any further; perform the following:

5. Type each command above in turn into the shell. You should see a red line on the display surface after you have performed the flip.
6. The line method also accepts an additional parameter at the end which determines the width of the line. Write a call of the line method that will draw a 5 pixels thick green line from 50,20 to 100,100. Type the statement in and prove that it works. Remember to flip so that you can see the results.
7. There is also a `circle` method which accepts the surface, the colour, the position, the radius and the line thickness.
8. Draw a blue circle at 100,100 of radius 50 and thickness 5 pixels.

Practical Break 3: Drawing loads of lines

In this break we are going to create a class which will draw a display of lots of lines. The class will be the container of a method that will actually do the work. The good news is that this means we will not have to keep typing things in, we will be compiling and running the program. This is the program we are going to create:

```
import pygame
import random

class DrawingFun:
    def drawLines(self):
        pygame.init()
        width=800
        height=600
        size = (width,height)
        surface = pygame.display.set_mode(size)
        for i in range(0,100):
            startx = random.randint(0,width)
            starty = random.randint(0,height)
            endx = random.randint(0,width)
            endy = random.randint(0,height)
            red = random.randint(0,255)
            green = random.randint(0,255)
            blue = random.randint(0,255)
            pygame.draw.line(surface, (red,green,blue), (startx,starty), (endx,endy),5)
        pygame.display.flip()

fun = DrawingFun()
fun.drawLines()
```



Before you go any further; perform the following:

9. Use **File>New File** in Idle to create a new program file.
10. Copy the program into the new file. It is quite a lot of typing, so you might want to find an electronic version of the text and use cut and paste to move the text around.
11. Run the program and admire the pretty coloured lines.

The program makes use of the random number facility in Python which will give integers in particular ranges. It picks a set of random start and end positions, along with a random colour, and

then draws a line of that colour on the screen. But you know that, you've seen it run. Of course you can now modify the program to make it do interesting things.



Before you go any further; perform the following:

12. See if you can make a method called `drawCircles` that draws circles of random size and colour all over the screen.
13. You can set the background colour of the screen to any colour you like:
`surface.fill((255,255,255))`
The fill method is given a tuple that sets the red, green and blue component of the background. The call above would fill the background with white (best to do this *before* you draw the lines. Change your program to draw the circles on a grey background.

You can also use the framework to draw images for you:

```
def drawCheese(self):
    pygame.init()
    width=800
    height=600
    size = (width,height)
    surface = pygame.display.set_mode(size)
    cheeseImage = pygame.image.load("cheese.png")
    cheeseRect = cheeseImage.get_rect()
    cheesePos = (40,60)
    surface.blit(cheeseImage, cheesePos)
    pygame.display.flip()
```

This method opens up an image from a file called `cheese.png` and then draws it on the screen. The file is loaded from the same folder that contains the program. You can find a cheese image [here](https://rob-miles.squarespace.com/s/cheese.png):

<https://rob-miles.squarespace.com/s/cheese.png>



Before you go any further; perform the following:

14. Follow the link above in your browser, right click the image and then save it with the filename `cheese.png` in the same location as your program file.
15. Draw some cheese using the method above.
16. Draw some cheese on a blue background.
17. Draw 100 randomly positioned cheeses.

Rob Miles June. 2014